BAB XI. Operasi File

Operasi dasar file pada prinsipnya terbagi menjadi 3 tahap, yaitu:
- membuka atau mengaktifkan file
- melaksanakan pemrosesan file
- menutup dile

A. Membuka file

Sebelum suatu file dapat diproses, file harus dibuka terlebih dahulu. Sebelum file dibuka, terlebih dahulu obyek file harus didefinisikan. Sintaksnya:

```
ofstream nama_obyek;
```

perintah ofstream dapat dijalankan dengan menyertakan file header **fstream.h**

Setelah itu, suatu file dapat dibuka dengan perintah

```
nama_obyek.open("nama file dan path");
```

B. Menulis ke File

Salah satu jenis pemrosesan pada file adalah menulis atau merekam data ke file. Sintaknya:

```
nama_obyek << ... ;
```

C. Menutup File

Setelah pemrosesan file selesai, file dapat ditutup menggunakan perintah

```
nama_obyek.close();
```

Contoh 1.
Program berikut ini untuk menulis teks ke dalam file

```
#include<iostream.h>
#include<fstream.h>

void main()
{
 ofstream fileteks;
 fileteks.open("C:/algo.txt");

 fileteks << "Untuk mencapai tujuan yg besar, maka tujuan itu"
          << endl;
 fileteks << "harus dibagi-bagi menjadi tujuan kecil"<< endl;
 fileteks << "sampai tujuan itu merupakan tujuan yg dapat "
          << "dicapai" << endl;
 fileteks << "berdasarkan kondisi dan potensi yg dimiliki saat "
          << itu " << endl;

 fileteks.close();
}
```

perintah fileteks.open("C:/algo.txt"); akan membuka file algo.txt yang ada di C:\
Apabila file tersebut belum ada maka akan dibuat secara otomatis, dan apabila sudah ada isi file algo.txt akan terhapus.

D. Menambah Data pada File

Suatu file yang sudah ada sebelumnya dapat ditambah data yang baru (tidak menghapus data lama). Caranya dengan menambahkan perintah **ios::app** pada open().

```
nama_obyek.open("nama file", ios::app);
```

Contoh 2.

```
#include<iostream.h>
#include<fstream.h>

void main()
{
 ofstream fileteks;
 fileteks.open("C:/algo.txt", ios::app);

 fileteks << endl;
 fileteks << "Oleh: Al Khowarizmi << endl;

 fileteks.close();
}
```

E. Memeriksa Keberhasilan Operasi File

Tidak selamanya jalan yang mulus ditemui. Ada kemungkinan terjadi saat file dibuka, ternyata file tidak ada. Dalam C++ tersedia function untuk memeriksa kondisi-kondisi pada operasi file, sehingga kesalahan saat eksekusi dapat dikendalikan.

Function yang dimaksud adalah **fail()**.

Contoh 3:

```
#include<iostream.h>
#include<fstream.h>

void main()
{
 ifstream fileteks; { ifstream digunakan u/ membaca file }
 fileteks.open("C:/algo.txt");
 if (fileteks.fail()) cout << "Maaf file takdapat dibuka/"
                           << "tidak ditemukan";
 fileteks.close();
}
```

F. Operasi Berbasis Karakter

Operasi file dapat dilakukan dalam bentuk karakter. Misalnya proses penyimpanan data ke file dilakukan setiap karakter, atau membaca data file karakter per karakter. Operasi ini didukung oleh function **put()** dan **get()**.

Contoh 4:

Program untuk menyimpan data karakter per karakter ke dalam file.

```
#include<iostream.h>
#include<fstream.h>

void main()
{
 ofstream fileteks;
 fileteks.open("C:/contoh.txt");
 fileteks.put('A');
 fileteks.put('B');
 fileteks.put('C');
 fileteks.close();
}
```

Contoh 5.
Program untuk membaca file karakter per karakter

```
#include<iostream.h>
#include<fstream.h>

void main()
{
  char karakter;
  ifstream fileteks; {}
  fileteks.open("C:/contoh.txt");

  while(!fileteks.eof())
  {
    fileteks.get(karakter);
    cout << karakter;
  }

  fileteks.close();
}
```

Latihan.

1. Buatlah program C++ untuk menghitung jumlah karakter dalam suatu file. Inputnya adalah nama file dan pathnya.
2. Buatlah program C++ untuk menghitung jumlah karakter tertentu, misalnya karakter 'A'. Input berupa nama file dan karakter yang akan dihitung.
3. Misalkan suatu file teks berisi listing program C++. Buatlah program untuk menghitung pasangan kurung kurawal yang ada pada file teks tersebut.
4. Buatlah program C++ untuk melakukan enkripsi shift chiper suatu file teks (dengan asumsi semua karakter huruf adalah huruf kapital). Inputnya adalah file teks yang akan dienkripsi dan besar pergeseran (integer). Outputnya adalah file teks hasil enkripsi.

Hint:
Ide dasar shift chiper adalah mengubah setiap karakter huruf ke karakter huruf lain. Misalkan pergeserannya 2, maka berikut ini karakter hasil enkripsi

awal	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
hasil	C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

Sehingga misal diberikan suatu teks C++ IS EASY, maka hasil enkripsinya adalah E++ KU GCUA